2007

Classification of Microarray Data to Predict Toxic Exposure

Tarek A. Seleem
Wright State University

Classification of Microarray Data to Predict Toxic Exposure

A thesis submitted in partial fulfillment
of the requirements for the degree of
Master of Science

By

Tarek A. Seleem
B. S., Cairo University, 1988

2007
Wright State University

WRIGHT STATE UNIVERSITY
SCHOOL OF GRADUATE STUDIES

September 11, 2007

I HEREBY RECOMMEND THAT THE THESIS PREPARED UNDER MY SUPERVISION BY Tarek A. Seleem ENTITLED Classification of Microarray Data to Predict Toxic Exposure BE ACCEPTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF Master of Science.

_____
Mateen Rizki, Ph.D.
Thesis Director

_____
Thomas Sudkamp, Ph.D.
Department Chair

Committee on
Final Examination

_____
Mateen Rizki, Ph.D.

_____
Thomas Sudkamp, Ph.D.

_____
T.K. Prasad, Ph.D.

_____
Joseph F. Thomas, Jr., Ph.D.
Dean, School of Graduate Studies

ABSTRACT

Seleem, Tarek A. M.S., Department of Computer Science, Wright State University, 2007. Classification of Microarray Data to Predict Toxic Exposure.

This thesis presents a software system for the analysis of microarray data. Microarrays are a relatively new technology that can be used to examine the state of the genome of an organism at some instant in time. The challenge is the amount of natural variation in biological systems limits our ability to identify specific genes that may be sensitive to changes in an organism's physiology or its environment. The analysis software consists of three modules. The first module filters microarray data to reduce the complexity of the problem. The second module selects subsets of genes for evaluation using a genetic algorithm. The final module uses a neural network to evaluate the selected genes to predict an organism's level of exposure to toxic substance. Results are present for a data set consisting of subjects exposed to eight levels of α-naphthylisothiocyanate (ANIT), a model hepatotoxin causing liver damage in the form of intrahepatic cholestasis.

TABLE OF CONTENTS

Page

LIST OF FIGURES

Figure Page

LIST OF TABLES

Table Page

I. INTRODUCTION

The objective of this thesis is to create a software system capable of analyzing microarray data. A microarray or gene-chip is a sensor that is used to take a snapshot of the state of the organism's genome at some instance in time. Each gene-chip records the activity of thousands of genes in the form of real-valued responses. By examining the gene-chips of healthy and diseased individuals, researchers hope to identify the specific genes that provide an early indication of pending health problems. Gene-chips can also be used to identify toxicological exposure. When organisms are exposed to biological or chemical agents, they exhibit changes in various biological pathways that are controlled by their genes. If these changes are recognized quickly, the individual or the contaminant can be removed from the environment perhaps reducing the harmful effects.

Gene-chip or DNA microarray technology was invented in the late 1980's by team of scientists at Affyxmetrix* and was made available to the scientific community in 1996. To effectively exploit gene-chip data, it is necessary to develop software tools that can be used to compare the gene-chips of different types of individuals (e.g. healthy vs diseased, unexposed vs. exposed, etc.). The problem that arises is typically there is a vast number of genes (tens of thousands) and relatively few data samples (dozens) available for analysis. This coupled with the large amount of naturally occurring biological variability in most organisms' genetic response makes it very challenging to find a small set of genes (biomarkers) that exhibit significant change relative to a specific health or environmental event.

The software system presented in this thesis includes three basic components: a filter module that reduces the number of genes (features) available for forming subsets, a gene selection module that assembles small sets of genes and a classification module to label samples based on activity of the selected set of genes. The software is evaluated using a data set provided by the Toxicology Branch at Wright-Patterson Air Force Base (AFRL / HEPB). This data set consists of 29 samples of rat liver that have been exposed to non-lethal doses of α-naphthylisothiocyanate (ANIT). These samples represent 8 levels of toxic exposures ranging from 0 mg to 100 mg. The specific goal is to find a minimal set of genes (biomarkers) that is sufficient to accurately classify the level of exposure of these samples because the Air Force wants to build biosensors and the current technology will not support sensors that incorporate large numbers of genes. The specific approach used in this thesis is outlined in Figure 1.

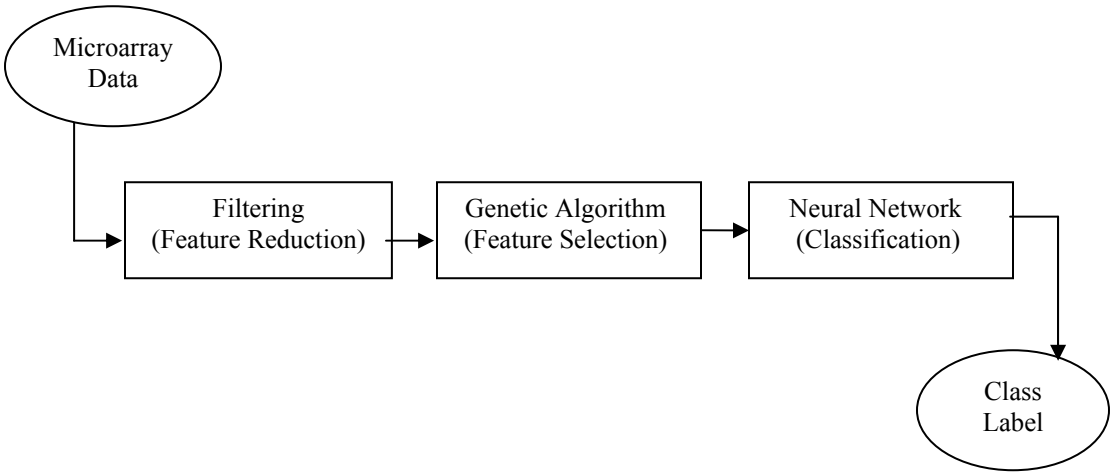


Fig. 1: Diagram of the Model Main Components

A collection of filter criteria are used to reduce the size of the set of features by an order of magnitude. A genetic algorithm is then used to assemble small sets of genes for evaluation. Finally, a neural network is used to evaluate the features and associate labels with each sample.

Background

1. Microarray Technology

A microarray is composed of a collection of microscopic DNA spots attached to a solid surface, such as glass, plastic or a silicon chip forming an array for the purpose of monitoring expression levels for thousands of genes simultaneously. Fluorescent tags are chemically attached to the strands of DNA. The tags or spot will then fluoresce (or glow) when examined. The intensity of the glow indicates the level of activity of genes under a particular condition. In addition, for each intensity value a reliability measure referred to as a p-value is computed. The p-value is a statistical measure that indicates the probability (ranging from zero to one) of whether the observed results in an experiment could have occurred by chance. Small p-values indicate that it is very unlikely that the results were due to chance.

2. Feature Reduction Techniques

Feature reduction techniques are used to reduce the number of features used in the classification process while retaining sufficient information to accurately discriminate among classes [2]. Typically a feature reduction process utilizes some mathematical method such as singular value decomposition (SVD) or principal component analysis (PCA) to produce a meaningful subset of features that retains the most useful information for classification.

Feature reduction has been used in various pattern classification problems where it is essential to reduce the computational complexity of classification system. It is meant to minimize the size of the input data vector by eliminating any redundant features and outliers (false data points caused by human or experimental errors). Finally, and most importantly, it helps to identify the patterns, features or characteristics that best represent the original experimental data.

3. Genetic Algorithms

The Genetic Algorithm (GA) [4] is a method of creating or evolving solutions for optimization problems by means of simulated evolution. The process is based on principles of natural selection. A population of individuals, which represents potential solutions, are artificially mated to produce new solutions that are evaluated and selected for survival based on their performance. Over time, the number of above-average individuals increases, hopefully producing a good solution to the problem.

In GA crossover is an operator that combines information from two or more parental individuals to produce a new individual referred to as an offspring. There are many types of crossover. Some techniques focus on random combinations of the parental information while others attempt to direct the choice of portions of the parental information that are combined. Ultimately crossover is a mechanism that generates sample points in the space of all possible solutions for a given problem. The crossover operator is limited to searching areas of the solution space defined by some combination of existing parental solutions. Unfortunately there may be areas of the solution space that cannot be reached by combining existing solutions so a mutation operator is also included. The mutation operator generates a new point in the search space without the restrictions imposed by the crossover operator. A mutation factor is used that controls the amount of variation allowed as the result of mutation. A large mutation factor will tend to increase the diversity of the population of candidate solutions, but may slow convergence towards a final solution.

4. Artificial Neural Networks

Artificial neural networks (ANN) are a collection of mathematical techniques that are used to solve a variety of problems in the areas of pattern recognition, process control, and signal filtering. In this thesis ANNs are used to define a classifier that assigns class labels to samples of data. Typically the internal structure of a neural network is collection of layers of nodes, connection weights and activation functions. Once the user picks certain aspects of the internal structure (i.e. the number of layers and types of activation functions) a supervised learning process can be applied to adjust the

connection weight so the ANN functions as a classification system. In this thesis we used three types of neural networks, multilayered feed forward neural networks trained with back propagation [3], radial basis function networks [6] and probablistic neural networks [9]. Artificial neural networks have been used to classify microarray data. For example, in [1, 5] ANNs are used for cancer detection while other researchers have used ANNs to predict the efficiency of drugs [7, 8].

* Affymetrix is a leading bioinformatic company founded by Stephen P.A. Fodor, headquarters in Santa Clara, California, United States.

II. APPROACH

The goal of this project is to find a minimal set of genes that is sufficient to train a neural network so that it can accurately classify a set of microarrays. To solve the classification problem, we first need to filter the number of genes to reduce the complexity of the data set. Our target was to reduce the size of the data set to approximately 10% of its original size. To achieve this goal we applied feature reduction techniques to remove less useful genes. This in turn reduces the size of the search space for the genetic algorithm.

Data and the Classification Problem

Our microarray data consists of two 15866 x 29 matrices that represent measurements of 15866 genes for 29 subjects. The 29 subjects are divided into 8 levels of toxic exposure to the chemical agent α-naphthylisothiocyanate (ANIT) as shown in Table 1. Samples of the format of our data are shown in Tables 2 and 3. Specifically, Table 2 shows the intensities of the first 23 genes, while Table 3 shows the corresponding p-values.

Class Label	Level of exposure(mg)	Number of samples
1	**0.0**	**6**
2	**0.1**	**3**
3	**0.5**	**2**
4	**1.0**	**3**
5	**10**	**3**
6	**20**	**4**
7	**50**	**4**
8	**100**	**4**

Table 1: The 29 samples grouped by level of exposure

This problem is challenging because the number of samples is very small compared to the number of features. Thus, basic statistical techniques for data analysis are of limited value. It is also difficult to apply supervised learning techniques to this data. Typically, supervised learning requires sufficient data to define a training set to adjust the learning algorithm and a test data set to evaluate the generalization of the solution. Twenty nine samples are insufficient for training and testing a neural network classifier for such a large number of features when grouped in eight classes with some classes having as few as 2 samples.

Table 2: Sample of the Microarray or Gene-Chip Intensity Matrix Data

_____ 1 - 29 Samples in 8 levels of exposure (Classes) _____

classes	0.0						0.1			0.5		1.0			..
genes	1	2	3	4	5	6	7	8	9	10	11	12	13	14	..
1	1808.2	1741.8	1799.7	1257.1	1522.0	1732.0	1739.1	1917.7	1924.1	1661.2	1933.8	1640.0	1621.1	1752.2	..
2	1982.8	1969.0	2046.9	1775.2	1817.3	1774.2	1962.3	1984.2	1816.9	1964.5	1958.4	1891.0	1910.7	2015.1	..
3	1173.5	1268.7	1201.8	1215.5	1257.1	1187.3	1204.9	1329.1	1425.9	1435.6	1146.7	1328.7	1515.7	1223.2	..
4	3523.7	3538.9	3513.3	4106.9	3900.1	3920.0	3492.7	3661.2	2959.8	3916.2	3041.0	3980.9	3942.7	3989.8	..
5	4474.7	4620.9	4623.8	3726.2	4687.4	4624.0	4643.2	4832.4	4634.6	4904.9	5648.6	4837.3	4622.8	4888.9	..
6	1420.0	684.3	1477.1	1856.8	783.4	758.9	1593.4	1840.8	864.4	2123.8	1111.8	1325.0	1353.7	695.0	..
7	882.5	921.3	913.3	1120.9	768.7	931.4	940.9	1123.3	748.2	1101.1	897.3	925.1	1087.8	778.2	..
8	6765.3	6710.5	6871.9	4887.6	6695.5	6107.5	6997.7	8474.4	7635.6	8321.1	7402.3	5961.3	7297.3	7218.4	..
9	2985.3	2989.1	3098.8	3139.0	3098.2	3129.3	3086.7	3159.1	3202.4	3013.1	3147.1	3255.8	3361.8	3250.9	..
10	1722.1	1714.6	1700.0	1789.4	1292.0	1674.3	1709.7	2180.3	1893.6	1790.0	1956.5	1540.9	1799.4	1581.8	..
11	3110.0	3147.8	3188.0	2150.9	2556.2	2856.5	3128.7	2670.4	1865.1	2790.4	1891.4	2644.6	2752.1	2903.3	..
12	2682.1	2728.4	2703.0	2436.9	2500.1	2212.5	2701.6	3336.3	3582.9	2649.6	2496.6	2539.7	2502.9	2883.2	..
13	761.2	656.7	743.6	639.8	627.4	701.1	751.5	693.9	622.0	595.5	847.5	837.3	683.8	758.9	..
14	3200.7	3278.2	3258.0	2710.0	2451.9	2730.3	3337.7	3012.8	2477.5	2718.0	2565.9	2942.3	2367.2	2557.3	..
15	1553.4	1766.2	1712.1	1519.3	2017.8	1760.6	1568.9	1739.3	1299.6	1612.4	1844.6	1758.8	1476.8	2054.7	..
16	4977.6	4829.1	4890.3	3878.9	3313.4	2790.9	4826.6	3723.0	3696.4	4520.6	3624.9	4589.5	4874.4	4653.1	..
17	1167.4	1182.0	1198.0	953.4	731.6	1110.9	1134.4	919.6	1265.9	1031.4	985.1	1146.3	1015.1	1125.0	..
18	4762.0	4759.5	4852.9	3962.4	4866.4	4110.0	4910.0	3200.9	4942.6	4001.4	5806.0	4543.2	3447.5	4248.1	..
19	2140.4	2208.5	2273.6	2628.2	2269.2	2104.3	2163.0	3088.4	3035.1	3102.9	3174.2	2546.6	2413.7	2597.4	..
20	900.3	897.9	915.4	822.0	648.4	802.5	919.8	717.4	932.5	698.8	738.1	862.2	731.9	761.9	..
21	2112.3	2134.9	2133.8	2229.0	2436.6	2259.2	2145.4	2149.4	1952.1	2265.8	2146.8	1911.4	2108.3	2146.1	..
22	1170.6	1134.0	1110.8	987.7	1155.5	1119.2	1177.3	1010.7	1228.6	984.9	1115.6	1113.6	1083.8	1082.0	..
23	1538.3	1632.1	1502.6	1399.8	1392.9	1617.1	1269.8	1689.9	1687.7	1619.1	1425.6	1681.3	1658.0	1506.0	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..
15866	..	..	..	..	..	..	..	..	..	..	..	..	..	..	..

Table 3: Sample of the Microarray or Gene-Chip p-value Matrix Data

_____1 - 29 Samples in 8 levels of exposure (Classes) _____

classes	0.0						0.1		
genes	1	2	3	4	5	6	7	8	..
1	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
2	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
3	0.0012210	0.0007320	0.0012210	0.0007320	0.0007320	0.0007320	0.0012210	0.0012210	..
4	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
5	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
6	0.0041500	0.0012210	0.0041500	0.0107420	0.0041500	0.0029300	0.0041500	0.0107420	..
7	0.0080570	0.0080570	0.0080570	0.0141600	0.0080570	0.0080570	0.0080570	0.0058590	..
8	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
9	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
10	0.0007320	0.0007320	0.0007320	0.0012210	0.0002440	0.0007320	0.0007320	0.0007320	..
11	0.0029300	0.0029300	0.0029300	0.0029300	0.0029300	0.0029300	0.0029300	0.0029300	..
12	0.0012210	0.0012210	0.0012210	0.0019530	0.0012210	0.0012210	0.0012210	0.0012210	..
13	0.0002440	0.0002440	0.0002440	0.0019530	0.0002440	0.0007320	0.0002440	0.0012210	..
14	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
15	0.0019530	0.0007320	0.0019530	0.0029300	0.0019530	0.0012210	0.0019530	0.0007320	..
16	0.0012210	0.0012210	0.0019530	0.0058590	0.0012210	0.0019530	0.0012210	0.0058590	..
17	0.0007320	0.0012210	0.0007320	0.0019530	0.0058590	0.0029300	0.0012210	0.0058590	..
18	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
19	0.0007320	0.0007320	0.0007320	0.0058590	0.0007320	0.0007320	0.0007320	0.0007320	..
20	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0007320	..
21	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
22	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	0.0002440	..
23	0.0080570	0.0058590	0.0058590	0.0185550	0.0058590	0.0107420	0.0058590	0.0029300	..
..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..
..	..	..	..	..	..	..	..	..	..
15866	..	..	..	..	..	..	..	..	..

Analysis of the data revealed a large variation in the response of individual genes across samples and within classes with the same level of toxic exposure. We found that the intensity values across all samples range between (0) minimum – (44140.6) maximum. For example, the response of gene 1 for control subjects 1 and 4 is 1808.2 and 1257.1 respectively. While the response for gene 105 for subjects 4 and 6 is 17844.0 and 34576.0. These differences are of an order of magnitude.

The relationship between the response and the level of exposure is clearly not linear. The gene activity does not necessarily increase (or decrease) as toxic exposure increases. Figures 2 and 3 demonstrate the types of differences in gene behavior as a function of toxic exposure. For instance, examining the intensity values for gene 5335 (see Fig 2) at 0.5 mg, 1.0 mg and 50 mg of exposure (classes 3, 4 and 7) is quite different than the behavior of gene 13728 (see Fig 3). For gene 5335 we can see that the intensity goes up-up-down while the intensity of gene 13728 goes up-down-up.

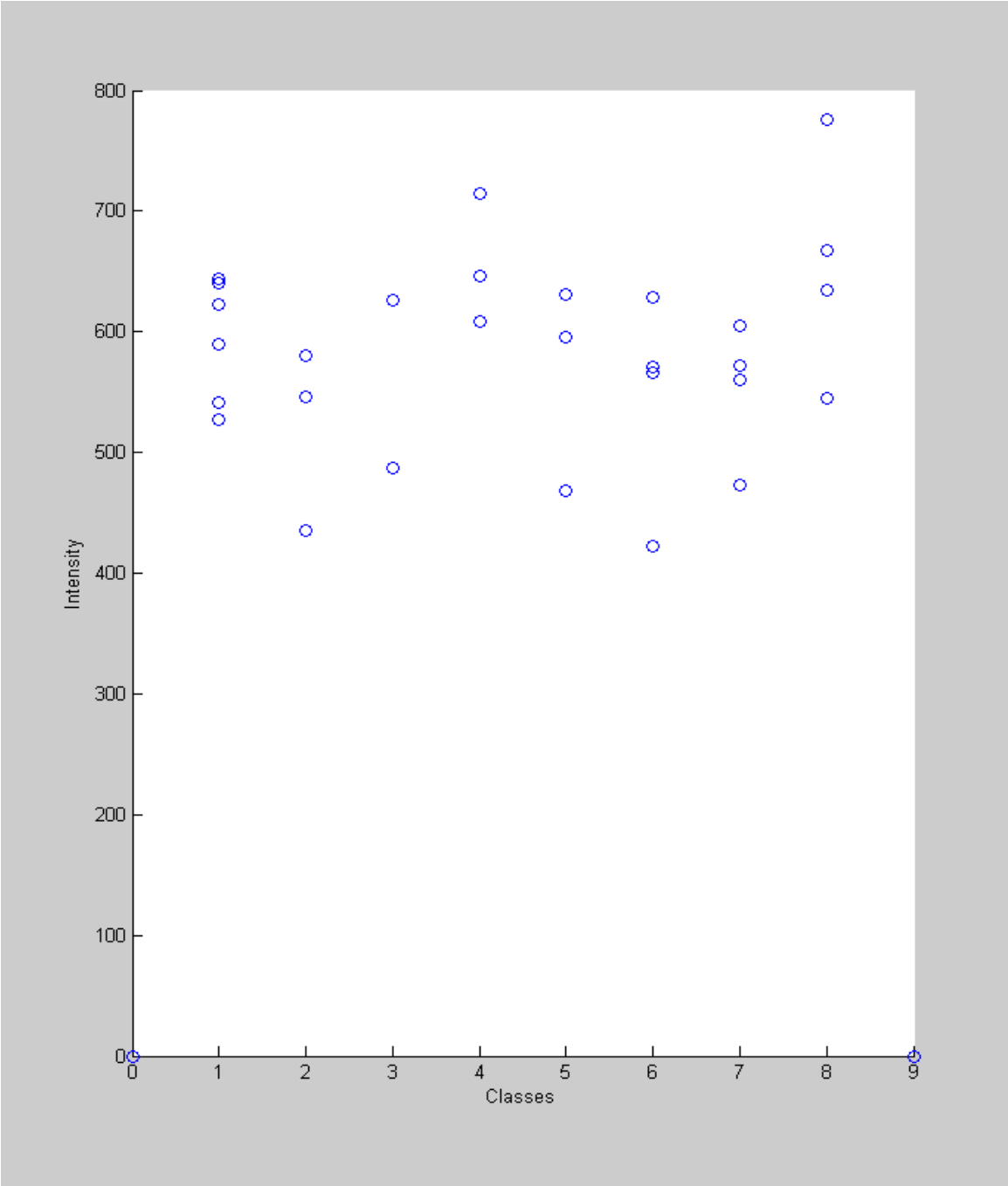


Fig 2: Intensity values for microarray gene [5335]

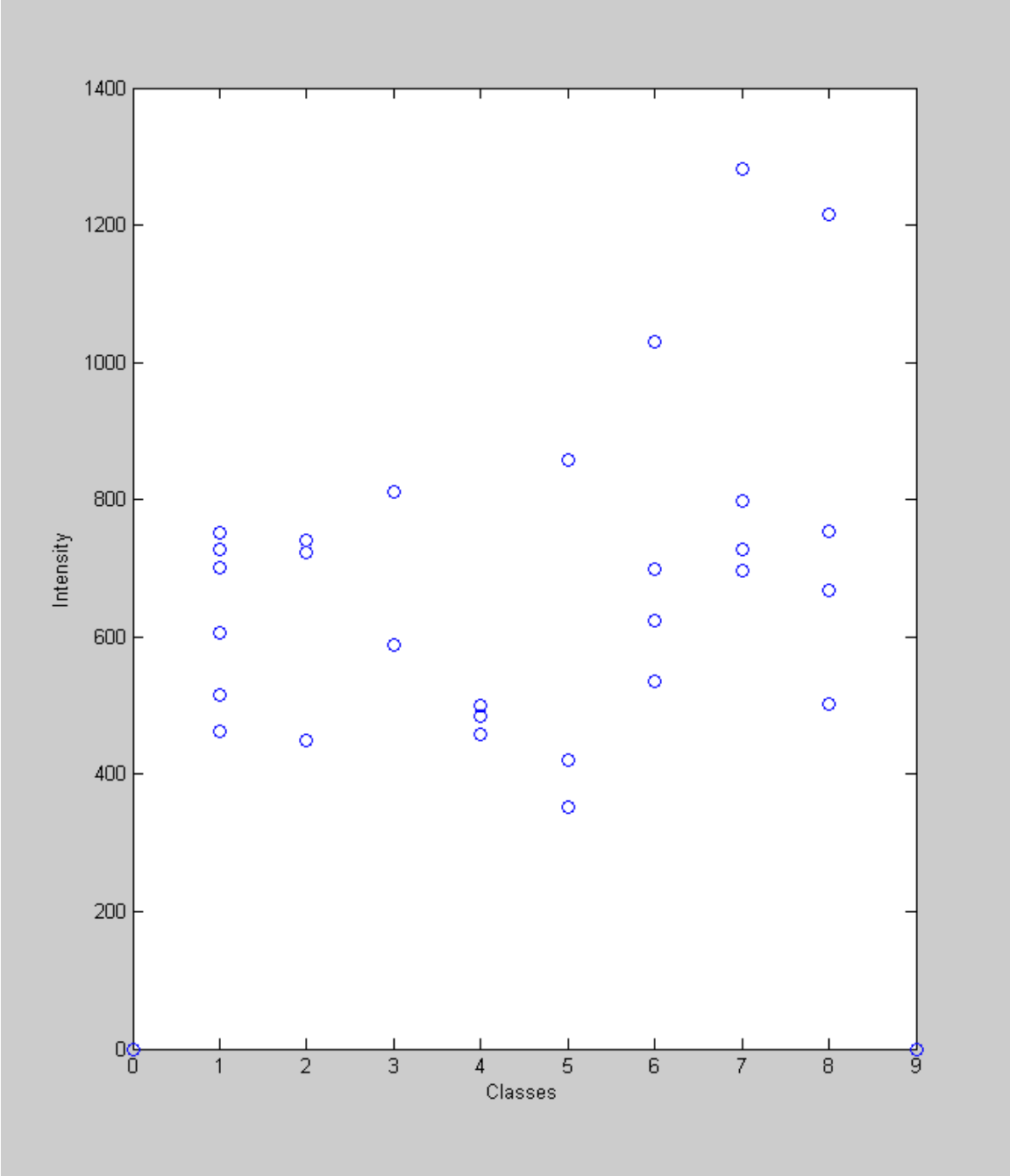


Fig 3: Intensity values for microarray gene [13728]

Feature Reduction / Filtering the Microarray Genes

In order to reduce the size of the gene pool by at least 90% (see Fig 4) we evaluated two algorithms for feature reduction.

Feature Reduction Technique 1

Technique 1 actually consists of two parts, the first part is applied to the p-value matrix and the second is applied to the intensity matrix. Each part selects a subset of features. In the final step we intersect the two subsets to get the final set of genes (FS-1). The common convention used by biologists in bioinformatics research is that if p-value (the probability that the result has occurred by chance) is greater than ($\Theta p = 0.05$) then it is considered unreliable. Hence for the first part of the technique, we considered only the genes with p-value across all samples less than or equal to Θp and discarded all the genes with p-values > Θp.

For the second part, we are interested in genes with considerable change in the intensity value across classes which means the exposed classes have a higher (or lower) mean value than the control class (0 mg of exposure). We applied two thresholds to the gene intensities to reduce the data set. First, we required the variance of a gene's mean intensities across the eight classes to exceed a threshold value ($\Theta v1$). This insures that there is a noticeable change (higher or lower than the average) in class response to the different level of exposure. Second, to reduce the effect of outliers (for error or an exceptional sample) we only kept genes that had a variance across their class-mean below a threshold value ($\Theta v2$). We chose these

($\Theta v1$) and ($\Theta v2$) as the upper and lower limits by empirically examining the effect of adjusting the various thresholds to achieve a reasonable size of data set. In other word, we considered genes whose variance lies in a certain window and have a higher (or lower) than average class-mean.

The pseudo code for the first part of the first filtering algorithm that produces gene set 1 (GS-1) is shown below.

(a) Select all genes with all their p-values < Θp

```
start with the first gene (row)
for each gene (row) do
    if any reading (column) > Θp then    // considered error
         go to next gene (row)
    endif
    save the position (index) of this gene
 end for-loop
 return the array of indices
```

The pseudo code used to produce gene set 2 (GS-2) is given below.

(b) applied to intensity matrix

```
start with the first gene (row)
for each gene (row) do
    if Θv2 > VAR(class-mean) > Θv1    // window considered
       save the position (index) of this gene
    endif
```

```
end for-loop
return the array of indices
```

The empirically selected thresholds used in this work for the first filtering algorithm are given in Table 4.

What it is	*variable*	value
p-value cut-off	Θp	0.05
Class-mean variance lower limit	$\Theta v1$	3200
Class-mean variance upper limit	$\Theta v2$	9600

Table 4: FS-1 Filter Parameters

After applying the two stages of the first filtering, the final reduced set of genes (FS-1) is created by intersecting GS-1 and GS-2.

FS-1 = GS-1 ∩ GS-2

This set we considered the "interesting" genes (features) that have sufficient responses to the toxic exposure for further analysis.

Feature Reduction Technique 2

Unlike the first filtering technique, the second technique only uses the p-value matrix. In this technique, in addition to the conventional cut-off p-value ($>\Theta p$), we

introduced a rather unconventional approach to reduce the number of genes while at the same time adding a very small chance that the data selected contains some random variation that might compensate for conditions that created variation in the experimental setup. We adjusted the feature selection criterion to consider the variance of the p-values. We computed the variance of the p-value for each gene, and set a very small threshold value ($\Theta v = 0.00002$) to be the minimum accepted variance over all the 29 samples. This means we discarded genes that had very similar p-value across all samples. The idea was to favor genes with intensity reading that were possibly the result of chance. We thought this added condition would alter the how the neural network was trained and ultimately improve generalization.

The second technique consists of two steps:

- First the algorithm checks all samples for each gene and discards any genes with any p-value greater than (Θp).
- Then, computes the variance of the p-value across the 29 samples for each gene and if the variance is less than (Θv) the gene is discarded.

The pseudo code for the second filtering technique is shown below.

```
start with the first gene (row)
for each gene (row) do
    if any reading (column) > Θp then    // considered error
       go to next gene (row)
    else
       if the variance (columns) < Θv  // not favorable
          go to next gene (row)
       end-if
```

```
        end-if
        save the pos. (index) of this gene
    end for-loop
    return the array of indices
```

The empirically selected thresholds used for the second filtering algorithm are given in Table 4.

What it is	*variable*	value
p-value cut-off	Θp	0.05
p-value variance over all 29 experiments	Θv	0.00002

Table 5: FS-2 Filter Parameters

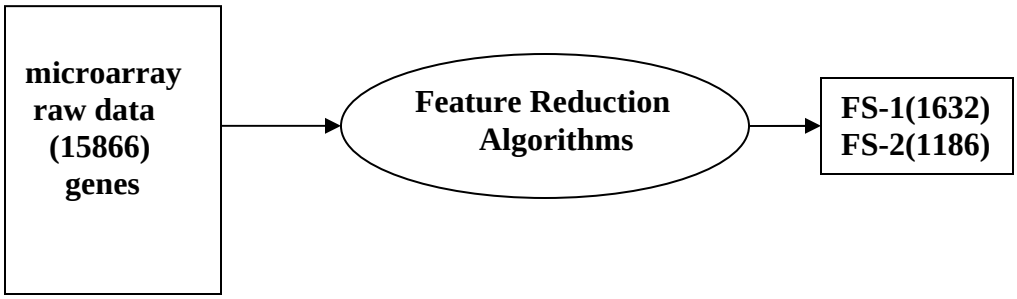


Fig 4: The microarray vector of (15866) genes filtered to (1632) and (1186), dimensionality was reduced to only about 10% and 7% respectively

Data Normalization:

The last step in data pre-processing was data normalization. This is an important step for the neural network. Normalization maps the input data to the interval [0, 1]. It is done by dividing the microarray intensity matrix values by its largest (max) data value in the entire array. Each set of filtered features of the microarray (FS-1 and FS-2) was normalized.

Genetic Algorithm and Feature Selection

A genetic algorithm (GA) was used for the feature selection process. The GA operates on a population composed of gene sets. Each gene set is represented as a vector of gene indices that refer to specific genes located in gene pools FS-1 or FS-2 generated using the filtering techniques. The vector of gene indices is referred to as a chromosome in GA terminology. The set of intensity values associated with the genes stored in a given chromosome are evaluated using a neural network. The percentage of data samples successfully classified by the neural network is considered to be the chromosome's performance (fitness). Individuals are sorted and only the fittest individuals survive and proceed to the next generation.

To find the smallest subset of genes that is sufficient to train the neural network, the genetic algorithm was run using various sizes of chromosomes. The system was tested with a chromosome size of 15 and if the accuracy of the best gene set was 100% a new run was initiated with a smaller chromosome size.

GA Initial Population:

Starting with the normalized gene sets (FS-1or FS-2) we randomly form the initial population of 400 chromosomes. For instance, since we chose the starting number of genes in a chromosome as 15, we reshaped the data available after filtering (FS-1 or FS-2) to randomly form a matrix of size 400x15 gene's indices as initial population.

After evaluating each chromosome's performance (fitness) we then sorted the population and then repeatedly selected pairs of chromosome to mate. We randomly

chose two individuals and applied the crossover operator to form the offspring and then mutated the offspring to form the new individual. We generated 2000 offspring and then selected the fittest 397 from the offspring. We also retained the best 3 individuals from the previous generation to form the initial population of 400 as the next generation.

The following pseudo code describes the flow of the genetic algorithm:

```
choose initial population (400 chromosomes)
repeat (1000 generations)
    for 1:5 do (5x400 = 2000 chromosomes total)
       apply crossover operator at random
       apply mutate operator
       call ANN to classify
       evaluate each individual chromosome's fitness
    end-for
    sort and select best (397 + 3) individuals to survive
until terminating condition
```

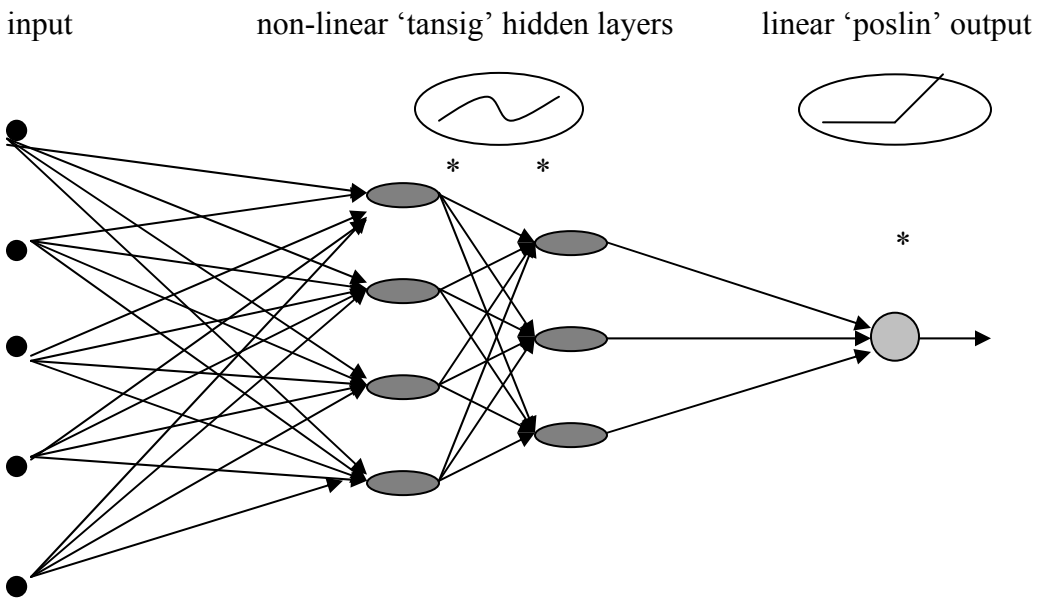
Classification and Evaluation

Once a new chromosome is formed we need to assess its performance. This is a measure of how well this chromosome (or set of features) performs as a classifier. We chose to use a neural network as the classifier.

We are trying to train our neural network to correctly classify a set of input genes values, which can be thought of as a learning concept. We then expect that when the neural network is presented with an unknown set of values associated with the same genes, it will tend to exhibit generalization by responding with a similar output.
This generalization property makes it possible to train a network on a representative set of input/output pairs and then use the resultant network for new data.
We have used the following three types of artificial neural networks:

1. Feed forward back-propagation network (FF-BP)

Feed-forward Back-propagation is a multiple-layer network with nonlinear differentiable transfer functions. Input vectors and the corresponding target vectors are used to train a network until it can approximate a function, associate input vectors with specific output vectors, or classify input vectors [3].

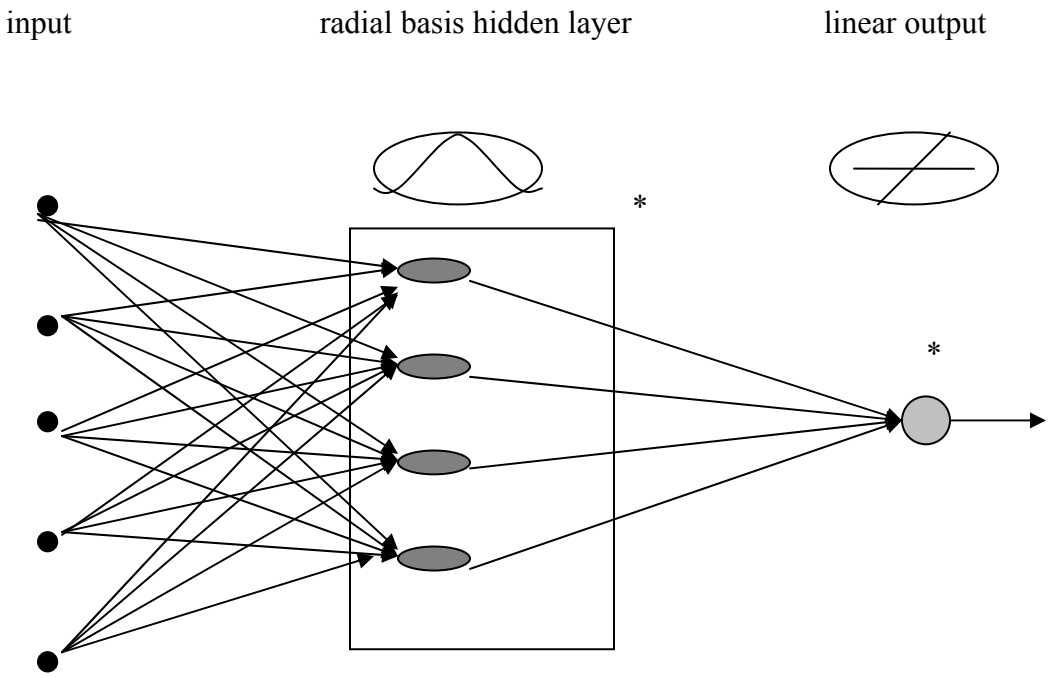


*number of neurons is for display only and is not the actual number in this layer.

Fig 5: Feed-Forward Back-Propagation Neural Network

2. Radial basis function network (RBF)

Radial basis networks consist of two layers: a hidden radial basis layer of neurons and an output linear layer. The radial basis functions layer computes the Euclidian distances for each input from the target class [6].

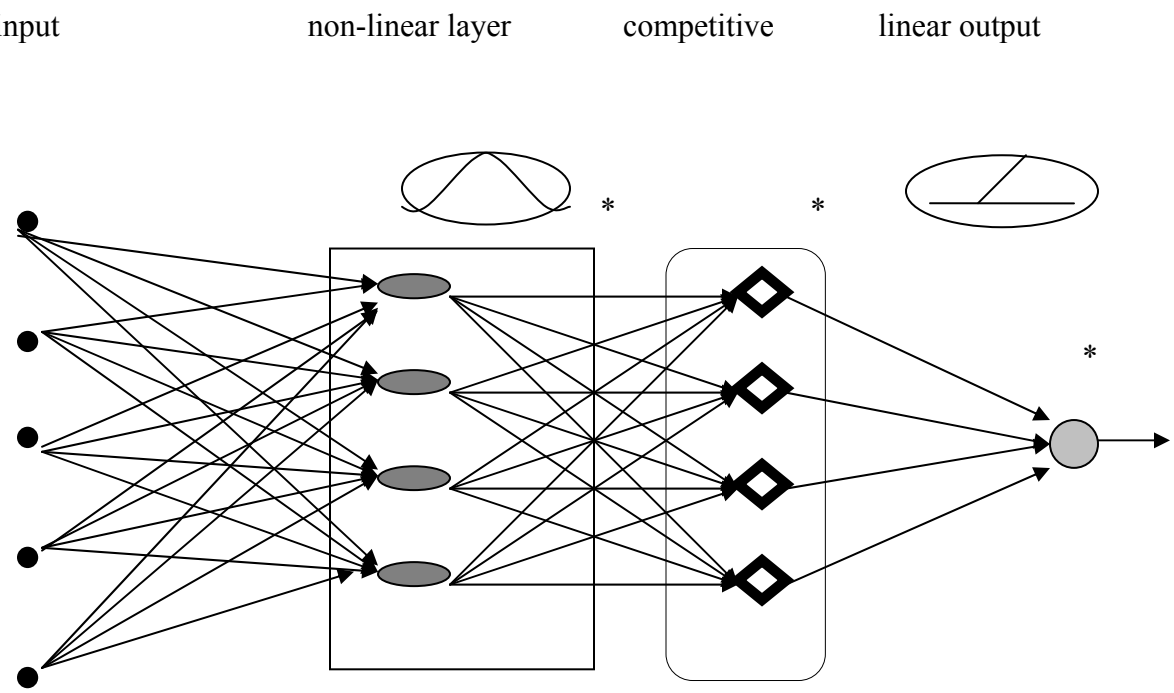


*number of neurons is for display only and is not the actual number in this layer.

Fig 6: Radial Basis Function Neural Network

3. Probabilistic network (PNN)

Probabilistic neural networks consist of two layers, the first layer computes distances from the input vector to the training input vectors, and produces a vector whose elements indicate how close the input is to a training input. The second layer sums these contributions for each class of inputs to produce as its output a vector of probabilities. Finally, a compete-transfer function on the output of the second layer picks the maximum of these probabilities, and produces a 1 for that class and a 0 for the other classes [9].



*number of neurons is for display only and is not the actual number in this layer.

Fig 7: Probabilistic Neural Network

III. EXPERIMENT AND RESULTS

Three experiments were performed using different neural network architectures. In the first experiment we used a feed-forward back-propagation neural network and feature set FS-2. In the second experiment we used the radial basis function neural network and feature set FS-2. In the third experiment we used the probabilistic neural network and feature sets FS-1, FS-2. As mentioned before, in all three experiments a genetic algorithm was used for the feature selection process while the neural network was used for evaluating the set of gene's (chromosome) classification performance hence assigning the fitness to the feature set.

Pre-processing the Microarray Data

Filtering the microarray genes

For dimensionality reduction, we applied the first filter technique to produce FS-1 using the parameters given in Table 4 and the second technique to produce FS-2 using the parameters shown in Table 5. The original data set contained 15866 genes. The FS-1 consisted of 6019 features selected using part (a) and 2299 features selected using part (b). Finally, we intersect the two subsets to produce the FS-1 set of 1632 features. Feature set FS-2 consisted of 1186 features.

Normalization of Input Data

We normalized the microarray intensity matrix data by using the maximum value of each feature set. For the FS-1 we divided the intensity matrix values by the FS-1 maximum value of 6751.9. For the FS-2 we divided the intensity matrix values by the FS-2 maximum value of 21073.

The Model

Figure 8 shows a flowchart for the entire system. We will discuss the system's main functions in more detail. First, we will describe the set-up of the genetic algorithm for feature selection process. Later we detail the set-up of the neural network experiments for the evaluation process.

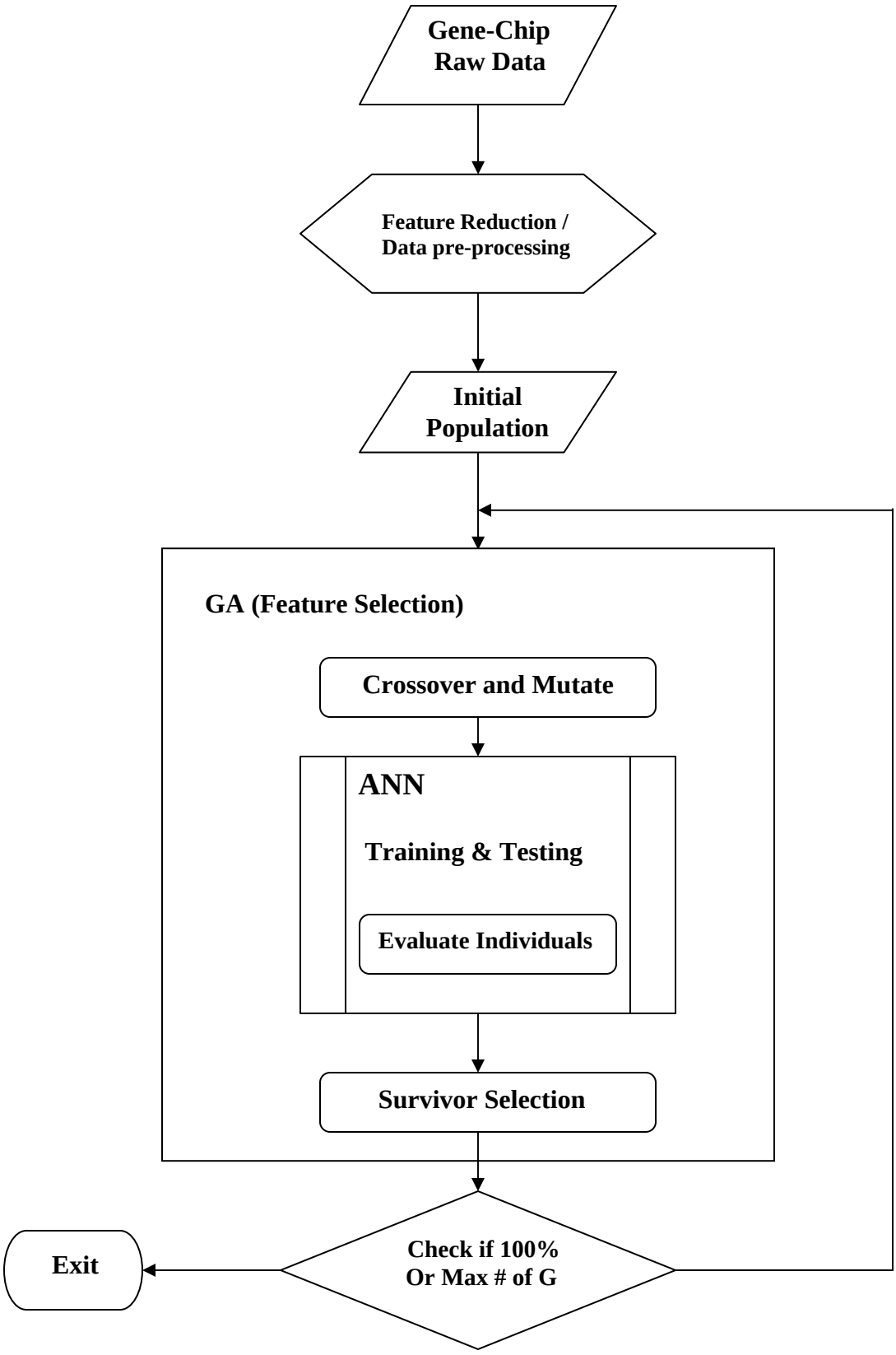


Fig 8: Model Diagram and Data Flowchart

The Genetic Algorithm Set-Up

The genetic algorithm was used in the model for the feature selection process. The set of features (chromosome) is evaluated by using the neural network as classifier. GA first applies crossover and mutate operators to individuals (chromosomes). It then calls the neural network to evaluate each chromosome's performance and sets the chromosome's fitness. After a population of 2000 chromosomes is evaluated, only top performing chromosomes survive. They are sorted and the best 397 are selected and added to the best 3 from old generation and the 400 individuals proceed to form the next generation. The process runs for 1000 generations or until a chromosome is found with 100% classification accuracy.

Number of generations per run	*1000*
Number of individual chromosomes per generation	*2000(exp2,3), 500(exp1)*
Number of fabricated data points for training per class	*10(exp2)*
Number of fabricated data points for testing per class	*5(exp3)*
Number of offspring per generation	*5(exp2,3), 1(exp1)*
Number of genes per chromosome	*started with 15 then10,9,8,7*

Table 6: GA Parameters Set-Up

GA Crossover

The crossover used in the genetic algorithm was a uniform crossover [10]. Individuals were randomly picked to mate, and for each gene in each parental chromosome there was a 50% chance that the gene index from the first parent are placed in the offspring's chromosome and a 50% chance the gene index of the second parent was placed in the offspring's chromosome.

GA Mutation

The mutation rate used was set to 25% in the first 50 generations of the run to prevent premature convergence. Then the mutation rate was reduced to 10% for the rest of the run.

GA Natural Selection For Survival

Natural selection is a very important step in the GA because it decides who will survive and proceed to next generation. A modified form of elite selection was used in this work. In experiment 1, FF-PB NN, individuals were sorted based on their score on the 21 samples for training and in case of a tie they were sorted based on their score on the remaining 8 samples. In experiment 2, RBF NN, individuals were sorted based on their score on 80 fabricated training samples. If there were ties, individuals were sorted based on their score on the 29 data samples. In experiment 3, PNN, first the extended population (parents + offspring) were sorted based on their score on the 80 fabricated training samples. If there were ties, these individuals were sorted based on their score on the fabricated 40 test samples. If there were still ties, then the tied individuals were sorted

based on their score on another fabricated set of 8 test samples. While 80 samples is created once in each run (1000 generations), the 40 samples and 8 samples were created by the NN for each evaluation. The 29 data samples were not used in selection process (GA) so it formed a true independent test set.

Neural Networks Set-Up

The following is the set-up for the different types of NN used in the experiments:

1. Feed-Forward Back-Propagation Neural Network:

In experiment (1) we used FF-BP network. We used 21 samples in eight classes as input for training and 8 samples for testing (one per class). The smaller feature set FS-2 was used to train. The system evolved for 1000 generations. The FF-BP was very slow and 100% classification was not achieved. We observed a tendency to over-fit the training data. In addition, the length of the chromosome did not seem to have any significant impact on the system's performance.

For training we used the microarray intensity data of 21 samples spanning all 8 classes. The samples used are [1, 2, 3, 4, 5, 7, 8, 10, 12, 13, 15, 16, 18, 19, 20, 22, 23, 24, 26, 27, 28]. The remaining 8 samples [6, 9, 11, 14, 17, 21, 25, 29], one from each class, were held back for testing. The feature set used in this experiment was the FS-2 since it is 28% smaller than FS-1. The set of experimental parameters is summarized in Table 7.

NN Results

The system was very slow (15 min per generation) despite the fact that each generation consisted of only 500 individuals. The system ran for 1000 generations and the best result achieved was 90% for training, 19 classes out of 21 were successfully classified; and it was 75% for testing, 6 classes out of 8 were successfully classified (see Table 10). There were concerns that the system was too customized to the training data.

Experiment 1

Component	*Variable*	*Value*	
Filter: *FS-2 (1186)*	*P-Value Θp*	*0.05*	
	P-Value Var. Θv	*0.00002*	
GA	*Initial Population Size*	*500*	
	Reproduction Size	*500*	
	Chromosome Size	*10*	
	Generations	*1000*	
NN	*Type*	*BP*	
	Num. Of Layers	*3*	
	Num. Of Nodes In Layer	*1*	*10*
		2	*5*
		3	*1*

Table 7: Experiment 1 Parameters Set-Up

2. Radial Basis Function Neural Network

In experiment (2) we used a RBF network. To reduce the possibility that the system was biased toward the 29 samples, we created a synthetic data set consisting of 10 samples for each class. Each sample was generated by randomly defining a value between the minimum and the maximum values observed in the real data for the given class. This produced (8 classes × 10 samples per class =) 80 values for each genes. We used the synthetic data for training and used the 29 real samples for testing. In case of a tie a synthetic set of 8 samples is created on the fly to use as a tie breaker in the GA selection process. The system evolved for 1000 generations. The RBF NN spread (σ) used was 0.03. The number of genes in each chromosome was set to 10. The feature set used in this experiment was the FS-2 since it is 28% smaller than FS-1. The set of experimental parameters is summarized in Table 8.

NN Results

This system was much faster than the FF-BP (3¾ min per generation) even though 2000 offspring were generated for each generation. The system ran for 1000 generations. The best result achieved was 100% for training, 80 samples out of 80 were successfully classified. It was 93% accurate on the test set, 27 samples out of 29 were successfully classified see Table 10.

While the training results were not biased to the sample data, there were concerns that the GA feature selection process was still biased to the real 29 data values.

Experiment 2

Component	*Variable*	*Value*	
Filter: *FS-2 (1186)*	*P-Value Θp*	*0.05*	
	P-Value Var. Θv	*0.00002*	
GA	*Initial Population Size*	*400*	
	Reproduction Size	*2000*	
	Chromosome Size	*10*	
	Generations	*1000*	
NN	*Type*	*RBF*	
	Num. Of Layers	*2*	
	Num. Of Nodes In Layer	*1*	*8*
		2	*1*

Table 8: Experiment 2 Parameters Set-Up

3. Probabilistic Neural Network

In experiment (3) we used a PNN network. We used the previously created synthetic 80 data values for training. To avoid the possibility that the GA was biased to the real 29 samples data in the feature selection process, we created another synthetic data set for testing. This synthetic test set contained 40 samples (5 samples per class) created using the same procedure described in the previous section. We resample the test set every generation, while the 80 training samples were generated at the beginning of the run. In the case of a tie a synthetic set of 8 samples was created and used as a tie breaker in the GA sorting process. We then used the 29 samples for a one-shot test at the end of the evolutionary process. Thus, the real data was not used in the GA sorting and survivor selection process. Both FS-1 and FS-2 were tested. The chromosome length started with 10 then was lowered to 9 for FS-1 and to 7 for FS-2. This network performed the best among all three types of neural networks; it correctly labeled all 29 samples with the minimum number of features (see Table 10). The set of experimental parameters is summarized in Table 9.

NN Results

At first, the feature-set used was the FS-2 since it is 28% smaller than FS-1 and 100% classification result was achieved after 128 generations. Then the feature-set FS-1 was used and again the model was able to achieve the 100% classification result after 343 generations.

PNN and FS-1

The system was slightly faster than the RBF (3½ min per generation) even though 2000 individuals were generated per generation. The system ran for 343 generations; the final results achieved were 100% for training, 80 samples out of 80 were successfully classified; 100% for testing with synthetic data, 40 samples out of 40 were successfully classified; and 100% for testing with experimental data, 29 samples out of 29 were successfully classified (see Table 10). The best chromosome contained 9 genes and 8 were unique.

PNN and FS-2

This system also was slightly faster than the RBF (3¼ min per generation). The system ran for 128 generations; the results achieved were 100% for training, 80 samples out of 80 were successfully classified; 100% for testing with synthetic data, 40 samples out of 40 were successfully classified; and 100% for testing with experimental data, 29 samples out of 29 were successfully classified (see Table 10). The best chromosome contained 7 genes and 5 were unique.

We think that the reason we have duplicated genes in the best chromosomes is because we used synthetic data for training. Consequently slightly different values would be entered in the NN during the evaluation process even if the gene indices are the same.

Experiment 3

Component	*Variable*	*Value*	
Filter: *FS-1 (1632)*	*P-Value Θp*	*0.05*	
	Mean Intensity Var. Θv1	*3200*	
	Mean Intensity Var. Θv2	*9600*	
Filter: *FS-2 (1186)*	*P-Value Θp*	*0.05*	
	P-Value Var. Θv	*0.00002*	
GA	*Initial Population Size*	*400*	
	Reproduction Size	*2000*	
	Chromosome Size	*FS-1*	*9*
		FS-2	*7*
	Generations	*1000*	
NN	*Type*	*PNN*	
	Num. Of Layers	*3*	
	Num. Of Nodes In Layer	*1*	*8*
		2	*8*
		3	*1*

Table 9: Experiment 3 Parameters Set-Up

Summary of Results

Two subsets of the original microarray genes were selected for training and testing the ANN. They are feature-set 1 (FS-1 of 1632 genes) and feature-set 2 (FS-2 of 1186 genes).

Feature set FS-1:

The FS-1 set consists of 1632 genes.

FF-BP

No test with this gene set since the 100% result was not achieved with FS-2.

RBF

No test with this gene set since the 100% result was not achieved with FS-2.

PNN

A chromosome was of length 9 genes was found containing 8 unique genes.

Best result achieved was 100% for training and 100% for testing see Figure 9.

chromosome = [2283 5110 5409 5700 7441 11098 14029 14464 14464]

To help read the model output data (Figures 9 and 10) columns left to right:

Error: shows specifically in which sample was the model prediction error.

Desired: the result (label) desired for this data sample.

Best: the model predicted result for that sample.

Population Prediction Avg.:

To display the average of all population predicted results for that sample.

This figure was used as an indicator of the speed of convergence for the model.

%100 - %100 - %100

==> %100 of data classified after 343 generations

Genes : 2283 5110 5409 5700 7441 11098 14029 14464 14464

Training80 fitness ====> 80/80
Synthatic40 fitness ====> 40/40
Actual29 fitness ====> 29/29

Error	Desired	Best	Population Prediction Avg.
0	1	1	1.635
0	1	1	1.98
0	1	1	1.7575
0	1	1	1.07
0	1	1	1.52
0	1	1	1.37
0	2	2	2.7275
0	2	2	2.18
0	2	2	2.0225
0	3	3	3.1525
0	3	3	3.17
0	4	4	3.925
0	4	4	4
0	4	4	3.9225
Low/High----------14/15----------			
0	5	5	5.005
0	5	5	4.725
0	5	5	4.985
0	6	6	5.8225
0	6	6	5.59
0	6	6	5.9275
0	6	6	5.7675
0	7	7	6.765
0	7	7	6.135
0	7	7	7.0625
0	7	7	6.845
0	8	8	7.6675
0	8	8	7.37
0	8	8	7.7625
0	8	8	7.89

Fig 9: Final output result for the FS-1 after 343 generations

Feature set FS-2:

The FS-2 consists of 1186 genes.

FF-BP

The chromosome consisted of 10 unique genes.

The best result achieved was 90% for training and 75% for testing.

RBF

The chromosome consisted of 10 unique genes.

The best result achieved was 100% for training and 93% for testing.

PNN

The length of the chromosome was 7 genes and 5 were unique.

The best result achieved was 100% for training and 100% for testing

see Figure 10.

chromosome = [2004 3248 5335 13175 13175 14464 14464]

%100 - %100 - %100

==> %100 of data classified after 128 generations

Genes : 2004 3248 5335 13175 13175 14464 14464

Training80 fitness ====> 80/80
Synthatic40 fitness ====> 40/40
Actual29 fitness ====> 29/29

Error	Desired	Best	Population Prediction Avg.
0	1	1	1
0	1	1	1.18
0	1	1	1
0	1	1	1
0	1	1	1
0	1	1	1
0	2	2	2.3
0	2	2	2.02
0	2	2	2.04
0	3	3	3
0	3	3	3
0	4	4	4
0	4	4	4.08
0	4	4	3.96
Low/High----------14/15----------			
0	5	5	5
0	5	5	4.1
0	5	5	5
0	6	6	6
0	6	6	6
0	6	6	6.22
0	6	6	6.02
0	7	7	6.64
0	7	7	7.04
0	7	7	7.12
0	7	7	6.16
0	8	8	8
0	8	8	7.9
0	8	8	8
0	8	8	8

Fig 10: Final output result for the FS-2 after 128 generations

Model result summary

ANN		training	testing	classified %		gene chromosome		generations	per gen. time in minutes
				train	test	# of genes	unique		
Back-Propagation	**FS-2***	**21**	**8**	**90**	**75**	**10**	**10**	**1,000**	**15****
Radial Basis Function	**FS-2**	**80**	**29**	**100**	**93**	**10**	**10**	**1,000**	**3¾**
Probabilistic	**FS-1 (1632)**	**80**	**40(&29)**	**100**	**100**	**9**	**8**	**343**	**3½**
	FS-2 (1186)	**80**	**40(&29)**	**100**	**100**	**7**	**5**	**128**	**3¼**

***FS-2 = Feature Set-2** ****population = 500**

Table 10: The neural network model final classification result

IV. CONCLUSION

The objective of this thesis is to create a software system capable of analyzing microarray data. The analysis software consists of three modules. The first module filters microarray data to reduce the complexity of the problem. The second module selects subsets of genes for evaluation using a genetic algorithm. The final module uses a neural network to evaluate the selected genes to predict an organism's level of exposure to toxic substance. We used a genetic algorithm to select a set of candidate genes that are evaluated using a neural network. Before solving the classification problem we filtered the genes to reduce the complexity of the problem.

Three experiments were conducted using different types of artificial neural networks: a feed-forward back-propagation network, a radial basis function network and a probabilistic neural network. While neural networks are very effective in classification problems, we discovered that the critical step in the system was the filtering process. The choice of filter strongly influenced the speed, structure and accuracy of the ANN results.

Two subsets of genes were found with 100% classification accuracy, hence one subset (solution) for each of the two filtered gene sets. A subset containing 9 genes (8 unique genes) was discovered for the first filtered feature set (FS-1) and a subset containing 7 genes (5 unique genes) was found for the second filtered feature set (FS-2).

Future Work

In most cases, with the proper pre-processing and system setup, it is possible to find a set of genes that achieves acceptable classification accuracy, but we have no way to know if these genes have any biological significance. It would be beneficial if the features set (genes) selected contained all the genes within related biological pathways. For future work, it will be beneficial if bioinformatics researchers help determine which feature sets are more biologically significant.

REFERENCES

[1] Bloom, G., Yang, I. V., Boulware, D., Kwong, K. Y., Coppola, D., Eschrich, S., Quackenbush, J. and Yeatman, T. J.,
"Multi-platform, multisite, microarray-based human tumor classification",
Am. J. Pathol. (2004) 164, pp. 9-16.

[2] Dash, M., Liu, H., "Feature selection for classification",
Intelligent Data Analysis, vol. 1 (1997), pp. 131–156

[3] Hecht-Nielsen, R., "Theory of the backpropagation neural networks", Proceedings of the international joint conference on neural networks, Washington DC, vol. 1. IEEE Press, New York (1989), pp. 593–605.

[4] HOLLAND , John H., "Genetic algorithms.",
Scientific American *Vol. 267* (1992), pp. 66-72.

[5] Khan J, Wei JS, Ringnér M, Saal LH, Ladanyi M, Westermann F, Berthold F, Schwab M, Antonescu CR, Peterson C, Meltzer PS,
"Classification and diagnostic prediction of cancers using gene expression profiling and artificial neural networks", Nat Med, Vol. 7, No. 6. (June 2001), pp. 673-679.

[6] Moody, J., Darken C. J., "Fast learning in networks of locally tuned processing units",
Neural Computation, vol.1 (1989), pp. 281-294.

[7] Natsoulis, G., El Ghaoui, L., Lanckriet, G. R. G., Tolley, A., Leroy, F., Dunlea, S., Eynon, B. P., Pearson, C., Tugendreich, S., and Jarnagin, K.,
"Classification of a large micro-array dataset. Algorithm comparison and analysis of drug signatures", Genome Research vol. 15 (2005), pp. 724–36.

[8] Sardari, S., Sardari, D.,
"Applications of artificial neural network in AIDS research and therapy", Current Pharmaceutical Design, vol. 8 (2002), pp. 659-670.

[9] Specht, D. F., "Probablistic Neural Networks",
Neural Networks, vol. 3 (1990), pp. 109-118.

[10] SYSWERDA, G. "*Uniform Crossover in Genetic Algorithms*," Proceedings of the 3 rd International Conference on Genetic Algorithms, (1989), pp. 2-9